

UNITED STATES PATENT APPLICATION

of

Brian Deen

Alex Hoppman

Joel Soderberg

Sean Lyndersay

for

**SENDING NOTIFICATION THROUGH A FIREWALL
OVER A COMPUTER NETWORK**

WORKMAN, NYDEGGER & SEELEY
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

FOR Filing

BACKGROUND OF THE INVENTION

1. The Field of the Invention

The present invention relates to methods and systems for sending event notification. More specifically, the present invention relates to methods and systems for determining an appropriate protocol to use when notifying a client computer inside a firewall of events that occur outside the firewall.

2. The Prior State of the Art

The popularity of the Internet has profoundly improved the way people communicate by allowing users quick and easy access to information. By accessing the World Wide Web and electronic mail through computers and other devices, people now stay in touch with each other around the globe, and can access information on a virtually limitless variety of subjects.

However, transmitting and storing information on the Internet raises various security issues. Any device using the Internet to store or transfer information is vulnerable to attack from all other devices on the Internet. As a result, many entities want the advantages of the Internet while still protecting their data and devices from attack. To meet the need for security on the Internet, a variety of products have been developed.

One product in particular is the "firewall." Firewalls are used to monitor communication between computer networks. If a firewall detects communication that might be a security risk, the firewall blocks the communication. Firewalls are often used to protect an entity's private network from exposure to the security risks inherent in communication over the Internet. In operation, a private network sits "inside" the firewall. When communication that might pose a security risk to the private network is detected

1 from a device "outside" the firewall, for example from the Internet, the communication is
2 blocked.

3 Firewalls are therefore advantageous for shielding a private network from harmful
4 communication originating on the Internet. Firewalls can be configured to block
5 communication based on various criteria, including content of the communication and
6 originating address of the communication. It may also be the case that a firewall is
7 configured to block content depending on the protocol that is used. For instance, when
8 using a connection-oriented protocol (like TCP), the firewall is configured to communicate
9 with clients "inside" the firewall that plan on sending data to, and receiving data from, a
10 machine "outside" the firewall. In this instance, the firewall acts more like a proxy server,
11 where the firewall establishes a connection with an outside server, on behalf of the client
12 machine, while protecting the identity of the internal machine from the external server.
13 However, when using a connectionless-protocol (like UDP) this security functionality is
14 disabled. Thus, many private networks use firewalls to block communication using
15 connectionless protocols in order to protect the private network.

16 However, a disadvantage of using firewalls to block communication is that
17 firewalls may inadvertently block useful communication, such as notification of the
18 occurrence of events, from entering onto a private network. For instance, a firewall
19 configured to block communications using a certain protocol will block all communication
20 using that protocol. It may be the case, that a device "outside" the firewall legitimately
21 needs to send communications to a device "inside" the firewall.

22 For example, in the context of event notification, the "outside" device may be
23 monitoring for the occurrence of an event that the "inside" device requested notification of.
24 If the monitored event occurs, the "outside" device may attempt to notify the "inside"

1 device of the occurrence. If a firewall blocks the protocol used by the “outside” device to
2 send the notification, the “inside” device is prevented from receiving the notification.
3 However, the “outside” device is unaware that the firewall is configured to blocked the
4 protocol and thus will continue to send notifications using the blocked protocol.

5 It is important with the ever-increasing number of users sending and receiving data
6 to devices on the Internet, that a device “inside” a firewall receives notification of the
7 occurrence of an event “outside” the firewall and that the notification is done as efficiently
8 as possible. Accordingly, methods and systems are desired for more efficiently notifying
9 devices “inside” a firewall of the occurrence of events “outside” the firewall.

SUMMARY OF THE INVENTION

The present invention relates to methods and systems for a client system inside a firewall to determine whether a server system outside the firewall can notify the client system of the occurrence of events. The client system and server system attempt to communicate using different protocols, until a protocol the firewall does not block is selected and the server system can contact the client system. Once contact is made, the characteristics of the selected protocol are utilized to send notifications to the client system in an efficient manner.

When a server system outside a firewall needs to notify a client system inside the firewall of the occurrence of an event, it would often be of benefit to ensure notification will take place and will be performed in an efficient manner. Therefore, when it is determined that a server system outside a firewall may have to notify the client system inside the firewall of the occurrence of an event, the determination is followed by the client system performing a series of acts, which ensure the client system will be efficiently notified using a protocol the firewall does not block. In absence of these acts, the client system may receive notification of the occurrence of the event in an inefficient manner or may never receive notification at all.

In operation, when a server system may have to notify a client system of the occurrence of an event, the client system determines the most efficient protocol to use to receive notifications. The client system first requests a UDP packet from the server system. If the client system receives a UDP packet from the server system in response to the request, the client system concludes that the server system is capable of notifying the client system on the occurrence of events using UDP. However, if the client system does not

1 receive a return UDP packet, the client system concludes the server system is unable to
2 notify the client system of the occurrence of events using UDP.

3 If communication using UDP failed, the client system would then rely on TCP to
4 receive event notifications from the server system. Due to the overhead associated with
5 using connection-oriented protocols, such as TCP, the client system would request that the
6 server system store data on the occurrence of events. The client system would then
7 establish a TCP connection to the server system at certain intervals and poll the server
8 system to see if any events occurred.

9 There is a twofold benefit to the current invention. First, the client system
10 determines the most efficient protocol it can use to receive notifications. The client system
11 first attempts to use UDP for receiving notifications. Since UDP is a connectionless
12 protocol, it requires less bandwidth and processor resources to transmit and receive data.
13 Only if the client system cannot receive communication using UDP does it then attempt
14 communication using the connection oriented TCP.

15 Second, upon selecting one of the protocols, either UDP or TCP, the invention uses
16 an efficient manner to receive notification based on the characteristics of the selected
17 protocol. When using UDP, the client system simply waits to receive a UDP packet
18 including the notification information. Since UDP is a connectionless protocol, having the
19 server system send a UDP packet, even repeatedly, when an event occurs uses minimal
20 Internet bandwidth and processor resources. On the other hand, the server system
21 repeatedly sending notification to the client system using the connection-oriented TCP
22 would consume significantly more Internet bandwidth and processor resources. As a
23 result, the notifications are stored so the client system can poll at certain intervals and
24 retrieve notification of the occurred events.

For the 4/29/60

1 Additional features and advantages of the invention will be set forth in the
2 description which follows, and in part will be obvious from the description, or may be
3 learned by the practice of the invention. The features and advantages of the invention may
4 be realized and obtained by means of the instruments and combinations particularly
5 pointed out in the appended claims. These and other features of the present invention will
6 become more fully apparent from the following description and appended claims, or may
7 be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the above recited and other advantages and features of the invention are obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof, which are illustrated, in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 illustrates an exemplary system that provides a suitable operating environment for the present invention;

Figure 2 illustrates some of the functional components present in a system where a client system may determine what type of protocol to use to receive notifications; and

Figure 3 is a flow diagram illustrating a method whereby a client system determines whether to use a connectionless or connection-oriented protocol to receive notifications.

DETAILED DESCRIPTION OF THE INVENTION

The present invention extends to both methods and systems for sending notification of the occurrence of events over a computer network where security devices on the network may block the transmission of notification data. The computer network includes at least one server system, one client system, and a communications link. The server system monitors events and the client system receives notification data associated with the occurrence of events. The notification data is received in a manner that prevents intermediate security devices from blocking the notification data's transmission. The embodiments of the present invention may comprise a special purpose or general-purpose computer including various computer hardware, as discussed in greater detail below.

The term "connectionless protocol" refers to protocols where a session is not established between two network devices before data transmission begins. Thus, there is no guarantee that the packets will get to the destination in the order they that were sent, or even at all. By way of example, and not limitation, User Datagram Protocol ("UDP") is a connectionless protocol.

In contrast, the term "connection-oriented protocol" refers to protocols where a session is established between two network devices before data transmission begins. Connection-oriented protocols often facilitate verification of the correct delivery of data between two network devices. Intermediate networks between the data's source and destination can cause data to be lost or out of order. Connection-oriented protocols correct transmission errors or lost data by detecting such errors or lost data and triggering retransmission until the data is correctly and completely received. Connection-oriented protocols also facilitate the correct reassembly of data packets even if the data packets have

1 arrived out of order. By way of example, and not limitation, Transmission Control
2 Protocol ("TCP") is a connection-oriented protocol.

3 The term "firewall" refers to a system designed to prevent unauthorized access to
4 or from a private network. Firewalls can be implemented in hardware, software, or a
5 combination of both. Firewalls are frequently used to prevent unauthorized Internet users
6 from accessing private networks connected to the Internet. All messages entering or
7 leaving the private network pass through the firewall, which examines each message and
8 blocks those that do not meet the specified security criteria. A device that is "inside" a
9 firewall is a device on the private network the firewall is preventing unauthorized access
10 to. A device that is "outside" a firewall is on a network the firewall is not preventing
11 unauthorized access to, such as the Internet.

12 Embodiments within the scope of the present invention also include computer-
13 readable media for carrying or having computer-executable instructions or data structures
14 stored thereon. Such computer-readable media can be any available media which can be
15 accessed by a general purpose or special purpose computer. By way of example, and not
16 limitation, such computer-readable media can comprise physical storage media such as
17 RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or
18 other magnetic storage devices, or any other medium which can be used to carry or store
19 desired program code means in the form of computer-executable instructions or data
20 structures and which can be accessed by a general purpose or special purpose computer.

21 When information is transferred or provided over a network or another
22 communications connection (either hardwired, wireless, or a combination of hardwired or
23 wireless) to a computer, the computer properly views the connection as a computer-
24 readable medium. Thus, any such a connection is properly termed a computer-readable

1 medium. Combinations of the above should also be included within the scope of
2 computer-readable media. Computer-executable instructions comprise, for example,
3 instructions and data which cause a general purpose computer, special purpose computer,
4 or special purpose processing device to perform a certain function or group of functions.

5 Figure 1 and the following discussion are intended to provide a brief, general
6 description of a suitable computing environment in which the invention may be
7 implemented. Although not required, the invention will be described in the general context
8 of computer-executable instructions, such as program modules, being executed by
9 computers in network environments. Generally, program modules include routines,
10 programs, objects, components, data structures, etc. that perform particular tasks or
11 implement particular abstract data types. Computer-executable instructions, associated
12 data structures, and program modules represent examples of the program code means for
13 executing steps of the methods disclosed herein. The sequence of instructions
14 implemented in a particular data structure or program module represents examples of
15 corresponding acts for implementing the functions or steps described herein.

16 Those skilled in the art will appreciate that the invention may be practiced in
17 network computing environments with many types of computer system configurations,
18 including personal computers, hand-held devices, multi-processor systems,
19 microprocessor-based or programmable consumer electronics, network PCs,
20 minicomputers, mainframe computers, and the like. The invention may also be practiced
21 in distributed computing environments where tasks are performed by local and remote
22 processing devices that are linked (either by hardwired links, wireless links, or by a
23 combination of hardwired or wireless links) through a communications network. In a
24

1 distributed computing environment, program modules may be located in both local and
2 remote memory storage devices.

3 With reference to Figure 1, an exemplary system for implementing the invention
4 includes a general purpose computing device in the form of a conventional computer 120,
5 including a processing unit 121, a system memory 122, and a system bus 123 that couples
6 various system components including the system memory 122 to the processing unit 121.
7 The system bus 123 may be any of several types of bus structures including a memory bus
8 or memory controller, a peripheral bus, and a local bus using any of a variety of bus
9 architectures. The system memory includes read only memory (ROM) 124 and random
10 access memory (RAM) 125. A basic input/output system (BIOS) 126, containing the basic
11 routines that help transfer information between elements within the computer 120, such as
12 during start-up, may be stored in ROM 124.

13 The computer 120 may also include a magnetic hard disk drive 127 for reading
14 from and writing to a magnetic hard disk 139, a magnetic disk drive 128 for reading from
15 or writing to a removable magnetic disk 129, and an optical disk drive 130 for reading
16 from or writing to removable optical disk 131 such as a CD-ROM or other optical media.
17 The magnetic hard disk drive 127, magnetic disk drive 128, and optical disk drive 130 are
18 connected to the system bus 123 by a hard disk drive interface 132, a magnetic disk drive-
19 interface 133, and an optical drive interface 134, respectively. The drives and their
20 associated computer-readable media provide nonvolatile storage of computer-executable
21 instructions, data structures, program modules and other data for the computer 120.
22 Although the exemplary environment described herein employs a magnetic hard disk 139,
23 a removable magnetic disk 129 and a removable optical disk 131, other types of computer
24

1 readable media for storing data can be used, including magnetic cassettes, flash memory
2 cards, digital video disks, Bernoulli cartridges, RAMs, ROMs, and the like.

3 Program code means comprising one or more program modules may be stored on
4 the hard disk 139, magnetic disk 129, optical disk 131, ROM 124 or RAM 125, including
5 an operating system 135, one or more application programs 136, other program modules
6 137, and program data 138. A user may enter commands and information into the
7 computer 120 through keyboard 140, pointing device 142, or other input devices (not
8 shown), such as a microphone, joy stick, game pad, satellite dish, scanner, or the like.
9 These and other input devices are often connected to the processing unit 121 through a
10 serial port interface 146 coupled to system bus 123. Alternatively, the input devices may
11 be connected by other interfaces, such as a parallel port, a game port or a universal serial
12 bus (USB). A monitor 147 or another display device is also connected to system bus 123
13 via an interface, such as video adapter 148. In addition to the monitor, personal computers
14 typically include other peripheral output devices (not shown), such as speakers and
15 printers.

16 The computer 120 may operate in a networked environment using logical
17 connections to one or more remote computers, such as remote computers 149a and 149b.
18 Remote computers 149a and 149b may each be another personal computer, a server, a
19 router, a network PC, a peer device or other common network node, and typically include
20 many or all of the elements described above relative to the computer 120, although only
21 memory storage devices 150a and 150b and their associated application programs 136a and
22 36b have been illustrated in Figure 1. The logical connections depicted in Figure 1 include
23 a local area network (LAN) 151 and a wide area network (WAN) 152 that are presented
24 here by way of example and not limitation. Such networking environments are

1 commonplace in office-wide or enterprise-wide computer networks, intranets and the
2 Internet.

3 When used in a LAN networking environment, the computer 120 is connected to
4 the local network 151 through a network interface or adapter 153. When used in a WAN
5 networking environment, the computer 120 may include a modem 154, a wireless link, or
6 other means for establishing communications over the wide area network 152, such as the
7 Internet. The modem 154, which may be internal or external, is connected to the system
8 bus 123 via the serial port interface 146. In a networked environment, program modules
9 depicted relative to the computer 120, or portions thereof, may be stored in the remote
10 memory storage device. It will be appreciated that the network connections shown are
11 exemplary and other means of establishing communications over wide area network 152
12 may be used.

13 In this description and in the following claims, a “computer” is defined as a general
14 purpose or special purpose computer or any other computing device including, but not
15 limited to, various computer hardware components such as those illustrated in Figure 1. A
16 “computer system” is defined as a group of one or more computers that interact to perform
17 one or more functions. A “client system” is defined as a computer system, group of
18 computer systems, other devices that might be associated with a network system, or
19 combination thereof, that use the services of another computer system. A “server system”
20 is defined as a computer system, group of computer systems, other devices that might be
21 associated with a network system, or combination thereof, that provide services to another
22 computer system. A “network system” is defined as a plurality of interconnected computer
23 systems and other network devices capable of being interconnected to computer systems.
24

1 Note that a computer system may use the services of another computer system and
2 yet still provide services to other computer systems. Thus, a client system in one context
3 may also be a server system in another context. Similarly, a server system in one context
4 may also be a client system in another context. This principal is applicable to all
5 embodiments of the present invention.

6 Figure 2 illustrates a network configuration suitable for implementing the
7 principles of the present invention. The configuration includes client system 210, private
8 network 220, firewall 230, public network system 240, and server system 250. Although
9 only one server system and one client system are illustrated in Figure 2, the general
10 principals disclosed herein can be readily adapted to configurations having any number of
11 client systems and server systems in combination. The private network system 220
12 includes the client 210 and is in communication with the firewall 230. The public network
13 system 240 includes the server 250 and is also in communication with the firewall 230.
14 Network configurations for private network system 220 include, but are not limited to,
15 Ethernet, token ring, Arcnet, or any other network configuration or combination thereof.
16 Public Network 240 can be any of these configurations, including the Internet.

17 Firewall 230 prevents communications from entering private network 220 based on
18 security criteria. For example, the firewall 230 may prohibit any UDP packets from
19 entering into the private network 220. The server 250 monitors for the occurrence of
20 events and may dispatch notification to the client 210 once a monitored event occurs. The
21 client 210 ideally receives the dispatched notification

22 In operation, client 210 requests, at some time before the notification of an event
23 might occur, that server 250 send client 210 a packet using a connectionless protocol.
24 Client 210 may make such a request when initially configured or when client 210 detects

1 that network configurations changed. For example, client 210 may request that the server
2 250 send a UDP packet. The request would pass over private network 220, through
3 firewall 230, over public network 240 and be received by server 250. While the client 210
4 is requesting a UDP packet in return, the protocol used to make the request is not limited to
5 UDP. Client system 220 can make the request using any protocol it is enabled to use,
6 including, but not limited to, TCP. After sending the request for a UDP packet, client 210
7 would then attempt to receive a UDP packet from the server 250.

8 Server system 250 receives the request and transmits UDP packet 251 to client 210.
9 UDP packet 251 would travel across public network system 240. However, when UDP
10 packet 251 reached firewall 230, firewall 230 would typically prevent the packet from
11 passing through and entering private network 220. Since client 210 will not receive UDP
12 packet 251, due to firewall 230 preventing it from entering private network 220, it is
13 determined that receiving notification using UDP is not viable.

14 This is advantageous because client 210 is made aware that it cannot receive
15 notifications using UDP before any notifications are actually sent. After client 210
16 determines it cannot receive notifications using UDP, it can request server 250, or any
17 other server, to use other methods (such as TCP) to notify client 210 of the occurrence of
18 an event. This reduces the chance that client 210 will not receive notification of the
19 occurrence of an event for which it has requested notification. In addition, the client 210
20 may poll the server 250 using TCP for event notifications rather than having the server 250
21 send notifications as they occur.

22 As stated above, it may be the case that client 210 sends a request for a packet
23 using a connectionless protocol when network configurations change. In Figure 2, if client
24 210 initially determined that it would receive event notifications using TCP, the removal

1 of firewall 230 might change the methods that are available. The discovery mechanism of
2 the current invention is completely dynamic, so that client 210 may receive notifications
3 using TCP when firewall 230 is attached to network 220. However when firewall 230 is
4 removed, client 210, without restarting, or performing any manual steps, re-connects to
5 server 250, and automatically shifts to receiving notifications using UDP.

6 The operation of the structure of Figure 2 will now be described with respect to
7 Figure 3, which is a flowchart of a client system operation when it is determined the client
8 is to receive notifications. The client begins by performing the step for determining if it
9 can receive communication from a server system using a connectionless protocol (step
10 306). In one embodiment, this may include the client requesting that the server send it a
11 packet of data using the connectionless protocol (act 301) and then attempting to receive a
12 packet of data from the server (act 302).

13 If the client does receive a packet of data from the server (YES in decision box
14 303), it is determined that receiving notifications can take place using the connectionless
15 protocol, such as UDP (act 304). In this embodiment, either there is no firewall or any
16 associated firewalls allow UDP packets to enter the private network. Since UDP packets
17 enter the private network, it is possible for the client to receive UDP packet notification.
18 Once the client receives a return UDP packet, it may request the server to send notification
19 of the occurrence of events using UDP. Alternatively, the server 250 may be configured to
20 send notification via UDP packets by default in which case no such express request would
21 be required. In the alternative case, the absence of an express request to use a protocol
22 other than UDP may be considered to be an implicit request to continue using UDP for
23 notification.

1 This embodiment has the advantage of using the connectionless protocol UDP to
2 send notifications. Since UDP is a connectionless protocol, even repeatedly notifying the
3 client of the occurrence of the same event takes minimal bandwidth and consumes minimal
4 processor resources.

5 However, if the client does not receive a packet of data from the server (NO in
6 decision box 303), for instance due to a firewall, it is determined that receiving
7 notifications should take place using a connection-oriented protocol, such as TCP (act
8 305). In one embodiment, where notifications are sent using TCP, a server stores data
9 associated with the occurrence of an event. Then, at time intervals, the client system polls
10 the server to access the data associated with the occurrence of the event. Since TCP is a
11 connection-oriented protocol, this polling method has the advantage of conserving
12 bandwidth on both associated public and private networks and on the processor resources
13 of the client and server systems.

14 If a client were to receive a TCP connection from a server on every recurrence of
15 every event, it would in some instances simply be notifying the client the occurrence
16 events the client is already aware of. Since the client is receiving redundant notifications
17 the verification algorithms and sequencing algorithms inherent in TCP consume bandwidth
18 and processor resources needlessly.

19 So whether a UDP packet is received by the client or not, the most efficient
20 protocol is used in the given situation and the characteristics of the chosen protocol are
21 used to deliver notification in a manner that preserves network bandwidth and processor
22 resources

23 The present invention may be embodied in other specific forms without departing
24 from its spirit or essential characteristics. The described embodiments are to be considered